

Listando os alunos

Transcrição

[00:00] Agora que nós já estamos persistindo esses alunos utilizando o Core Data, nós precisamos visualizar, nós precisamos conseguir editar, precisamos apagar.

[00:10] Só que para isso, o primeiro passo é mostrarmos esses alunos na table view da tela home, que é essa tela aqui. Até agora nós não a utilizamos, vamos começar a trabalhar nela agora. Como nós já persistimos um aluno, nós precisamos resgatar esse aluno e exibi-lo na table view. Então vamos lá?

[00:29] Eu vou fechar a pasta “Alunos” e vou abrir o View Controller da home, esse “HomeTableViewController”. Vamos pensar, nós precisamos obter os resultados dos alunos que o Core Data salvou pra nós. Pra isso vamos utilizar uma classe que se chama NS Fetched Results Controller, ela vai trazer os resultados pra nós.

[00:53] Para utilizar, o primeiro passo é importarmos o Core Data, então vou dar um “import CoreData”. Agora nós precisamos criar uma variável que nós consigamos acessar em todo o View Controller, porque nós vamos usá-lo em diferentes pontos. Por exemplo, vamos utilizá-lo para popular a table view, então o número de linhas que a tabela tiver é o número de alunos que esse gerenciador de resultados vai trazer para nós.

[01:17] Vamos precisar também resgatar o aluno aqui, no método “cellForRow”, para conseguirmos preencher a célula, o nome, a imagem e tudo mais. E também na hora que formos apagar um aluno, vamos utilizar essa lista de alunos. Para ficar mais fácil pra nós, vamos criar uma variável chamada “gerenciadorDeResultados”.

[01:42] Como eu tinha dito, ela é do tipo “NSFetchedResultsController”, esse cara aqui. Quando nós utilizamos essa classe, nós precisamos passar qual é a entidade que ele vai gerenciar pra nós. No caso é a “Aluno”, então vou passar aqui “Aluno” e deixá-la como optional pra nós a tratarmos embaixo.

[02:01] Feito isso, agora nós precisamos começar a configurar essa variável, entender como ela funciona. Vamos criar um método aqui? Que vai ser o “recuperaAluno”. Eu já vou chamá-lo aqui no “viewDidLoad” pra ele carregar logo quando a tela entrar, então “recuperaAluno”.

[02:27] Agora eu vou utilizar essa variável que nós criamos, o gerenciador de resultados. Então vou chamá-la aqui em cima, “gerenciadorDeResultados”, e eu vou inicializar essa variável para nós, ela é do tipo “NSFetchedResultsController”. Quando eu inicializo essa variável, ele nos pede um monte de parâmetros, que é o que nós estamos vendo aqui.

[02:51] O primeiro parâmetro é uma variável do tipo “NSFetchRequest”, que nós já vamos criá-la daqui a pouco. O segundo parâmetro é o contexto, mas parece que nós já tínhamos utilizado o contexto na hora de salvar o aluno, então nós podemos reaproveitar aquele cara. Esse “sectionNameKeyPath” nós não vamos utilizar e “cacheName” também não, porque nós não vamos trabalhar com cache na nossa aplicação, então nós não vamos utilizar esses dois.

[03:23] O primeiro parâmetro que ele pede é esse “NSFetchRequest”, então nós vamos criá-lo aqui, “let pesquisaAluno”, vou chamá-lo dessa forma. Ela vai ser do tipo “NSFetchRequest”, aqui eu preciso passar também qual é a entidade que eu quero que ele pesquise para nós, no caso é o “Aluno”.

[03:49] Eu tenho que inicializar essa variável pra nós começarmos a usar. Como eu faço pra inicializar? Eu chamo a entidade que nós queremos, no caso é “Aluno”, que nós temos lá, e eu dou um “.fetchRequest”, com isso ele já cria pra nós essa variável.

[04:06] Além disso, nós podemos utilizar esse “pesquisaAluno” pra setar outras configurações na busca. Como, por exemplo, se nós tivermos uma lista enorme de alunos, é legal que ordenemos, por exemplo, em ordem alfabética, fica mais fácil de acharmos o aluno. Então nós podemos setar essa configuração nessa variável “pesquisaAluno”.

[04:25] Primeiro nós temos que criar um Sort Descriptor, que é a variável que vai ordenar por nome para nós. Então vamos lá, “let ordenaPorNome”, ela é do tipo “NSSortDescriptor”. Eu vou inicializá-la, repare que o método construtor dela tem vários métodos construtores, o que nós vamos utilizar é esse aqui, em que nós vamos passar uma aqui e se é ascendente ou não, ou seja, se nós queremos que ordene em ordem alfabética ou não, então eu vou escolher esse método.

[05:01] Aqui em “key”, eu vou passar o nome, então vou pôr aqui “nome”. Esse “key” tem que ser algum dos atributos que nós colocamos no schema da nossa entidade aluno, então poderia ser endereço, poderia ser nome, poderia ser nota, apesar de que poderia ser, mas não faz muito sentido, por exemplo, por endereço. Mas o que eu quero dizer é que tem que ser um desses atributos do schema. No caso, nós vamos utilizar o nome.

[05:34] Como eu quero que ele ordene por ordem alfabética, de A a Z, eu vou passar aqui um “true”, porque eu quero que seja ascendente.

[05:43] Agora que nós já criamos o Sort Descriptor, nós precisamos setá-lo nesse “pesquisaAluno”, que é o “FetchRequest”, que é o cara que nós configuramos pra fazer a busca pra nós. Então eu vou dar aqui um “pesquisaAluno.”, aqui ele tem um “sortDescriptors”, então comecei a digitar, ele já preencheu aqui para mim.

[06:04] Repara que ele é um array, ele espera um array de Sort Descriptors, então o que eu tenho que fazer? Eu vou escolher esse cara, como ele espera um array, eu vou abrir aqui os colchetes e eu passo o Sort Descriptor que eu criei, que é esse “ordenaPorNome”. Poderia ter mais configurações aqui, como ele é um array, ele pode colocar mais de um.

[06:29] Já criei então o Fetch Request, que é o primeiro parâmetro que nós precisamos passar pro nosso gerenciador de resultados. Como eu já o criei, eu posso utilizá-lo aqui, então “pesquisaAluno”.

[06:42] O próximo atributo que nós temos que colocar aqui, é o contexto. Como eu já tinha dito, nós já utilizamos um contexto na hora de salvar o aluno, então eu posso aproveitar para utilizar o contexto, então eu vou copiar esse cara daqui e vou utilizar no nosso “HomeTableViewController”. Onde está “Variáveis” eu posso colocá-la aqui, assim nós conseguimos acessar. Aqui eu posso já passar o “contexto”. Aqui eu passo o “nil”, aqui também eu passo o “nil”, porque eu não vou utilizar.

[07:14] Então nós já montamos a inicialização da variável que nós criamos, que é o que vai trazer os resultados pra nós. Mas será que só isso já está funcionando, pessoal? Será que ele já está trazendo os resultados? Ainda não. O que nós precisamos fazer agora é fazer a operação realmente que o Core Data vai utilizar pra trazer esses resultados pra nós.

[07:35] O que eu preciso fazer? “gerenciadorDeResultados.performFetch”, que é esse método aqui. Repara que ele é um throw function, ou seja, ele exige que nós tratemos caso dê algum erro, pra isso nós precisamos colocar um “try”, então nós vamos verificar se isso vai funcionar.

[07:57] Além disso eu vou colocar um “do” e eu vou colocar esse “try” com nosso gerenciador de busca dentro do “do” e aqui eu coloco o “catch”, caso dê algum problema, nós vamos querer saber qual foi o erro, então eu posso pôr aqui “error.localizedDescription”. Legal, nós já tratamos. Com isso sim nós estamos conseguindo, nós estamos pedindo pro Core Data trazer os resultados pra nós.

[08:25] Só que se eu rodar o app agora, nós não vamos ver diferente, visualmente não vai acontecer nada. Embora estejamos chamando o método no “view DidLoad recuperaAluno”, visualmente não vai acontecer nada, porque nós

ainda não mexemos na table view, então ele está dando return 0, então já não vai nem cair aqui no método “cellForRow” e agora nós precisamos configurar a table view.

[08:51] O primeiro passo é trazermos de volta a lista de alunos, porque nesse método nós temos que falar para ele quantas linhas a table vai ter que renderizar, e a tabela vai ter que renderizar a quantidade de alunos que tiver persistindo no nosso device. Para fazer isso eu vou colocar “gerenciadorDeResultados.fetchedObjects”.

[09:13] Repare que aqui é um array, está vendo as chaves aqui de alunos? Só que ele é um optional, então se eu colocá-lo direto, desse jeito, “.count”, ele não vai aceitar, ele vai falar que ele espera um int e não um optional int, então nós precisamos tratar, desembulhar esse array.

[09:35] O que eu vou fazer? Eu vou fazer um “guard let” pra nós conseguirmos fazer isso. “guard let listaDeAlunos”, ele vai ser igual ao “gerenciadorDeResultados.fetchedObjects.count”, ele vai trazer pra mim a quantidade de alunos que tiver.

[09:54] Se ele não conseguir desembulhar, eu passo 0, porque provavelmente deu algum problema e ele nem vai cair mais no método cell for row, então não tem problema. Aqui no return eu posso apagar e agora eu passo a lista de alunos. Eu vou pôr aqui “contadorListaDeAlunos”, agora aqui ele está desembulhando com segurança, não tem problema. Agora nós precisamos mexer no método cell for row.

[10:27] Aqui nós precisamos, primeira coisa é trazer de volta a lista de alunos, então eu vou fazer o mesmo esquema, só que ao invés de eu pegar e trazer só o contador, eu vou trazer realmente toda a lista.

[10:37] Eu vou trazer aqui o “guard let”, como nós vimos, ele é um optional, então nós precisamos tratar isso, “lista”, na verdade aluno, porque eu vou pegar o aluno atual, então “aluno = gerenciadorDeResultados.fetchedObjects”, e eu preciso pegar o “indexPath.row”, porque se eu não colocar o “indexPath.row”, ele vai trazer toda a lista pra nós, nós não queremos toda a lista, nós queremos o aluno que ele estiver renderizando no momento de montar a table view.

[11:10] Se ele não conseguir desembulhar, ou seja, de ser ainda algum problema, eu vou dar um return da célula, então vou retornar célula. Agora sim nós podemos setar as informações que o Core Data trouxe para nós, através desse gerenciador de resultados, do fetched objects. Na célula, então, eu posso pegar “celula.labelNomeDoAluno.text”, eu posso pegar aqui o “aluno.nome”.

[11:41] Agora nós precisamos também colocar a foto. Como é que nós fazemos? Eu vou pegar a “celula.imageAluno.image”, vamos pegar o “aluno.foto”. Só que nós temos um problema aqui. Se eu colocar “aluno.foto”, repara que aqui é um “NSObject” e o que nós precisamos é uma UIImage, que é o tipo que nós esperamos pra setar nesse cara.

[12:07] Então aqui nós não podemos fazer isso, se nós colocarmos dessa forma, ele vai reclamar e não vai compilar, ele vai falar que NSObject não é do tipo UIImage, que é o que ele espera, então nós precisamos tratar isso daí.

[12:18] Como nós fazemos para convertê-lo novamente? Porque, na verdade, quando nós salvamos a imagem lá no Core Data, nós colocamos aquele transformable, pra conseguirmos transformar alguma coisa e salvar no Core Data, e depois nós precisamos fazer o casting de novo pro tipo que nós precisamos pra utilizar, que é o que eu vou fazer agora.

[12:36] Vou criar um “if let imagemDoAluno” é do tipo “aluno.foto”. Aqui eu vou dar um “as?”, que eu vou tentar transformá-la em “UIImage”. Se ele conseguir, ele vai entrar dentro desse “if let”, com toda a segurança, eu posso dar um “celula.imagemAluno.image” é igual a variável que nós criamos aqui no “if let”, então eu posso copiar esse cara e colocá-lo aqui. Eu vou tirar essa linha, agora nós estamos tratando certinho como nós fazemos pra setar a imagem na célula.

[13:19] Se eu rodar o app dessa forma, ainda a visualização da célula vai estar um pouco estranha, vamos rodar o app aqui só pra nós visualizarmos o problema. Então eu vou gerar um build.

[13:30] Tem dois alunos aqui que eu cadastrei, só que a célula está bem pequena, eu vou dar um zoom pra nós visualizarmos, o nome está cortado, a imagem está cortada. Isso aconteceu porque no storyboard ele está usando um tamanho, vamos ver qual é o tamanho? Vou clicar aqui na célula. Aqui, está 85. Só que não basta só nós colocarmos esse valor aqui, porque ele assume o valor default, então se eu tirar, olha como ele ficou, ele ficou igual está no simulador, então só isso aqui não vai rolar.

[14:05] Deixa eu pôr aqui 85 de novo, só pra nós conseguirmos visualizar essa alteração no storyboard, mas na prática só isso aqui não resolve, o que nós precisamos fazer é setar o tamanho da célula aqui, então eu vou utilizar o método “heightForRow”, é esse cara aqui. Agora sim eu posso dar um “return” do valor que eu quero para a célula, que é 85. Agora sim eu vou rodar o app novo pra nós conferirmos e vermos se ficou bonito.

[14:33] Bacana, agora sim nós setamos o tamanho correto da célula e nós estamos conseguindo utilizar o Core Data pra trazer esses resultados pra nós exibirmos os alunos que nós persistimos locais no nosso device.

[14:46] Vamos fazer outro teste? Vou vir aqui no “+”, vamos cadastrar mais um aluno. Vou vir em biblioteca, vamos pegar qualquer imagem, vou cadastrar o “Henrique”. Endereço, ele mora na Avenida Paulista, vou pôr “São Paulo”. Telefone. O site, “www.caelum.com.br” (<http://www.caelum.com.br%E2%80%9D>), a nota dele vai ser 7. Vou dar um salvar.

[15:21] Quando eu clicar em “salvar”, ele vai voltar para primeira tela, pra home. Só que não está aparecendo o Henrique, que é o que nós acabamos de cadastrar. O legal seria, eu entrei no formulário, cadastrei um aluno, ele retorna para a home e já com o novo aluno pra nós exibirmos. Então daqui a pouco nós vamos ver como nós fazemos pra solucionar esse problema.